

ILLINOIS INSTITUTE OF TECHNOLOGY



Big Data Technologies - CSP 554

Project Draft on Comparative Analysis of Cloud-Based NoSQL Databases

Team Members

Urjita Saxena	A20578349
Harsh Sharma	A20563703
Nikita Sharma	A20588827
Trushaliben Chandulal Tanti	A20598441

DEPARTMENT OF COMPUTER SCIENCE
COLLEGE OF COMPUTING

1. Review of the Literature

The explosion of big data applications has made scalable, schema-flexible, and high-throughput databases essential. Cloud-based NoSQL systems specifically Amazon DynamoDB (key-value/document store), MongoDB Atlas (document store), and Apache Cassandra/DataStax Astra DB (wide-column store) represent three distinct architectural approaches and CAP theorem trade-offs. This literature review compares them across architecture, data models, performance (individual and bulk operations), scalability, consistency, security, cost, benefits, and drawbacks, incorporating recent sources and benchmarks (2024-2025).

2. Architecture and Data Models

DynamoDB uses a serverless, ring-based architecture with automatic partitioning via partition keys (and optional sort keys) and built-in multi-AZ replication. It supports both key-value and document (JSON) models up to 400 KB per item.

MongoDB Atlas employs replica sets for HA and sharded clusters for horizontal scaling, storing data as flexible BSON documents (up to 16 MB) with rich nesting and dynamic schemas.

Cassandra/Astra DB follows a decentralized masterless ring topology (consistent hashing with vnodes), gossip protocol), using a wide-column (column-family) model optimized for massive write-heavy workloads.

3. Empirical Performance and Operational cost for Financial Scale

3.1 Performance - Individual Record Operations

DynamoDB consistently delivers single-digit millisecond latency (typically 4 to 10 ms) for point lookups and small writes, even at scale, with optional DAX for sub-millisecond reads (imsushant12, 2025; Knowi, 2025).

MongoDB Atlas offers good latency for complex queries but higher variability under heavy write loads due to WiredTiger storage engine overhead.

Cassandra excels in sub-2 ms write latency but reads can be slower unless data model is carefully denormalized.

3.2 Performance - Bulk Loading

Cassandra is the clear leader for bulk ingestion (billions of rows) via SSTable loading or tools like DSBulk, maintaining near-linear performance with minimal impact on online operations due to its LSM-tree design (Knowi, 2025).

DynamoDB handles bulk via parallel BatchWriteItem calls efficiently and cost-effectively in on-demand mode.

MongoDB bulkWrite() is capable but can be memory-intensive on the driver side and slower for very large imports compared to Cassandra.

3.3 Scalability and Consistency

DynamoDB: Serverless auto-scaling, strong consistency by default, seamless global tables which is best for predictable performance.

MongoDB Atlas: Automatic sharding, strong consistency within a replica set, eventual across shards which is excellent for query flexibility.

Cassandra/Astra DB: Near-linear horizontal scaling (add nodes, no downtime), tunable consistency (ONE/QUORUM/ALL) which is unmatched write scaling, multi-DC replication without single points of failure.

3.4 Security Features

All three offer encryption at rest or in-transit, VPC isolation, and auditing. DynamoDB integrates most tightly with AWS IAM and fine-grained access control. MongoDB Atlas provides field-level encryption and LDAP. Cassandra/Astra supports client-to-node encryption and role-based access but often requires more manual configuration in non-managed setups.

3.5 Cost Model (2025 pricing trends)

DynamoDB: On-demand is cheapest for spiky workloads; provisioned can be more economical for steady loads.

MongoDB Atlas: Cluster-based pricing (always-on cost).

Cassandra/Astra DB: Generally, most cost-effective at extreme scale when using serverless tiers.

4. Benefits and Drawbacks Summary

Database	Key Benefits	Main Drawbacks
DynamoDB	Lowest operational overhead, predictable latency, seamless AWS integration, serverless scaling	Vendor lock-in, limited query flexibility, can become expensive at very high sustained throughput
MongoDB Atlas	Rich querying, aggregation pipeline, multi-cloud, developer-friendly JSON documents	Higher memory footprint, sharding complexity, higher cost for always-on clusters
Cassandra/Astra	Unmatched write throughput & linear scaling, multi-DC resilience, no single point of failure	Steeper learning curve, requires denormalization, eventual consistency pitfalls if not tuned

While many sources provide detailed individual descriptions (e.g., vendor documentation and feature lists), recent independent benchmarks and comparisons are fewer and often outdated. This project addresses the gap by performing controlled head-to-head empirical testing (individual + bulk operations, provisioned vs on-demand, identical IoT/e-commerce datasets) to validate or challenge published claims (imsushant12, 2025; Bytebase, 2025; Knowi, 2025).

5. Contributions

This section contains the Team members Roles and Responsibilities:

S.No.	Contributions	Due Date	Owner
1	Finalize literature review & submit Project Draft	Nov 20, 2025	Urjita Saxena (lead), all review
2	Provision cloud environments (Atlas, DynamoDB, Astra DB)	Nov 25, 2025	Harsh Sharma
3	Dataset selection, cleaning & profiling	Nov 28, 2025	Nikita Sharma
4	Develop individual CRUD & query benchmark scripts	Dec 1, 2025	Trushaliben Tanti
5	Implement bulk loading & concurrent workload tests	Dec 4, 2025	Urjita Saxena + Harsh Sharma
6	Run experiments & collect metrics	Dec 7, 2025	All team members
7	Analysis, visualizations & draft results section	Dec 9, 2025	Nikita Sharma + Trushaliben Tanti
8	Final report (final submission)	Dec 10, 2025	All team members

6. References

- *imsushant12. (2025, Jan 27). Comparing Amazon DynamoDB with Other NoSQL Databases (MongoDB and Cassandra). <https://dev.to/imsushant12/comparing-amazon-dynamodb-with-other-nosql-databases-mongodb-and-cassandra-1h16>*
- *Bytebase. (2025, Apr 17). DynamoDB vs. MongoDB: a Complete Comparison in 2025. <https://www.bytebase.com/blog/dynamodb-vs-mongodb/>*
- *Knowi. (2025). Cassandra vs DynamoDB (2025): Complete Guide with Pricing, Performance & Migration Tips. <https://www.knowi.com/blog/cassandra-vs-dynamodb-2025-complete-guide-with-pricing-performance-usecase-migration-tips/>*
- *Han, J., et al. (2011). Survey on NoSQL Databases. Journal of Cloud Computing. <https://doi.org/10.1186/2192-113X-2-1>*